ON PRODUCT FORM OF INVERSES OF SPARSE MATRICES

R. P. Tewarson*

The Product Form of Inverse (1) is being used more and more often in large scale Linear Programming Systems. The inversion of large and sparse matrices is especially suited to the Product Form, provided such inverses are kept reasonably sparse (2, 3). Very few of the computational techniques, currently known, for keeping the Product Form of Inverse (PFI) sparse seem to be available in published literature (4, 5). In this paper, a geometrical interpretation of the PFI is utilized to explain the currently known techniques of keeping the PFI of a matrix sparse. Some proposed improvements in these techniques are also discussed.

Let A be a n x n non-singular matrix and I the associated identity matrix. The columns of I will be considered as a "basis" in a n-dimensional Euclidian space E_n. Each element of a given column of A matrix then is a coordinate of that column in terms of our basis. The columns of A can also be thought of as vectors in E_n centered at the origin. If a column s of A has an element $a_{rs} = 0$, then evidently column s is orthogonal to e_r - the unit basic vector (r^{th} column of I). Matrix A, which is sparse, can now be visualized as a matrix having most of its columns orthogonal to a large percentage of different basic vectors. In other words, the columns of a sparse matrix lie mostly in low degree hyperplanes in E_n.

[redacted]

* State University of New York at Stony Brook, N. Y. This research was supported by the National Aeronautics and Space Administration, Washington, D. C., Grant No. NGR-33-015-013.

Let us now discuss the well-known Gauss-Jordan method (6) and its relationship to PFI of A. We pick a column of A and find a linear transformation $T^{(1)}A = A^{(1)}$, where $A^{(1)}$ is a matrix with one unit column. Next, we find a $T^{(2)}$ so that $T^{(2)}A^{(1)} = A^{(2)}$ having two of its columns transformed to unit vectors. Proceeding in this manner $A^{(k)}$ will have k of its columns already transformed into unit vectors and finally $A^{(n)}$ will have all of its columns as unit vectors and it will be a permutation of I, say I_p. The n linear transformations of the type $T^{(k)}$ are such that

$$T^{(n)}\ T^{(n-1)}\ \ldots T^{(2)}T^{(1)}I_p = A^{-1}\ ,\ \text{the inverse of A.}$$

Let us take a closer look at $T^{(k)}$. Let $a_{ij}{}^{(k)}$ denote the element in row i and column j of the matrix $A^{(k)}$. Whenever a column of $A^{(k)}$ gets transformed into a unit vector we say it has gone into the basis and the original unit basic vector in that position is said to have gotten out of the basis. The linear transformation $T^{(k+1)}$ that transforms a non-basic column s of $A^{(k)}$ into a unit vector e_r is equivalent to the following operations. Divide the row r of $A^{(k)}$ by $a_{rs}{}^{(k)}$ and subtract multiples of this new row r from all the other rows of $A^{(k)}$ such that all the other elements of $a_{is}{}^{(k)}$ are reduced to zero. Evidently the preceding operations on $A^{(k)}$ are equivalent to the premultiplication of $A^{(k)}$ by a matrix that is obtained from I with its r^th column replaced by the transpose of the row vector.

$$\left(\frac{-a_{1s}{}^{(k)}}{a_{rs}{}^{(k)}}, \frac{-a_{2s}{}^{(k)}}{a_{rs}{}^{(k)}}, \ldots \frac{-a_{r-1,s}{}^{(k)}}{a_{rs}{}^{(k)}}, \frac{1}{a_{rs}{}^{(k)}}, \frac{-a_{r+1,s}{}^{(k)}}{a_{rs}{}^{(k)}}, \ldots \frac{-a_{ns}{}^{(k)}}{a_{rs}{}^{(k)}} \right)$$

Also, if $T^{(k+1)}$ is applied to e_r we get the above-mentioned column vector.

Such vectors are called eta vectors and the corresponding matrices are known as eta matrices. In other words, all the eta vectors are transformed representations (or coordinates) of original unit basic vectors at the instant they are displaced from the basis. We have seen that corresponding to each linear transformation $T^{(k)}$ there is an sta vector $\eta^{(k)}$. All of the n eta vectors constitute the PFI. In PFI the eta vectors once formed are not operated upon by subsequent transformations. In contrast, if the eta vectors are transformed we get the explicit inverse, this is the Gauss-Jordan method for matrix inversion. Thus, in computing PFI the number of columns to be transformed decreases by one at each stage, while in the evaluation of explicit inverse by Gauss-Jordan method all the columns are transformed at each stage.

To keep the PFI sparse the non-basic columns of $A^{(k)}$; $k = 0, 1, 2, \ldots (n - 1)$ are kept sparse because we have seen that the eta vectors are formed from non-basic columns of $A^{(k)}$' s. It will be shown below that by proper choice of vectors entering and leaving the basis, the growth of non-zero èlements in non-basic columns of $A^{(k)}$ can be controlled. The density $d^{(k)}$ of the matrix $A^{(k)}$ is defined to be the ratio of number of non-zero elements in the $(n - k)$ non-basic columns of $A^{(k)}$ to the total number of elements in those columns. Thus $d^{(0)}$ is the density of the original matrix $A^{(0)}$ or A. The density of PFI is equal to the total number of non-zero elements in all the eta vectors divided by n^2.

Let us introduce the row count vector $c^{(k)}$ and its significance at this stage (4). The number of non-zero elements in each row of the non-basic columns of $A^{(k)}$ is counted; the column vector of these n row counts is $c^{(k)}$. Therefore an element $c_i^{(k)}$, in row i of the row count vector, is the number of non-basic vectors that use the basic vector e_i

in their representations. Let $\min_{i} c_i^{(k)} = c_r^{(k)}$. Then e_r is the basic vector used in the representation of (one of) the least number of non-basic vectors or e_r is orthogonal to (one of the) largest number of non-basic vectors.

A linear transformation of the type $T^{(k+1)}$ (or eta matrix) that brings column s into the basis and removes unit basis vector e_r out of the basis will leave all non-basic columns of $A^{(k)}$ that are orthogonal to $e^{(r)}$ unchanged, because only those column vectors that have a non-zero element in row r will get changed if pre-multiplied by an eta matrix -- which we recall is an identity matrix with column r replaced by the eta vector. Therefore, in order that as many non-basic columns of $A^{(k+1)}$ be identical to the non-basic columns of $A^{(k)}$, we remove out of the possible original basic vectors that can be replaced by the given column vector, the one that corresponds to minimum $c_i^{(k)}$ (4). This procedure will evidently keep the growth of non-zero elements in non-basic columns of $A^{(k+1)}$ small.

We have thus far a criterion for deciding which original basic unit vector to remove from the basis at each stage, such that the density growth of the PFI is kept low. In order to determine which column of $A^{(k)}$ to bring into the basis at each stage, we proceed as follows. Define the density measure $D_j^{(k)}$ of each non-basic column vector $A^{(k)}$ as $D_j^{(k)} = \Sigma' c_i^{(k)}$, where Σ' denotes that the sum is taken for only the rows that correspond to non-zero elements of column j of the matrix $A^{(k)}$.

The non-basic column vector of $A^{(k)}$ having the minimum $D_j^{(k)}$ is chosen to enter the basis at step k. Thus min $D_j^{(k)} = D_s^{(k)}$, where $a_s^{(k)}$ is the column chosen to enter the basis. We can say that the hyperplane in which $a_s^{(k)}$ lies is not only of low degree but also is composed of basic vectors such that each of them is used in the representation of few non-basic vectors.

Columns of $A^{(0)}$ that have only one element (such columns are called singletons) when brought into the basis, leave the density of non-basic columns of $A^{(0)}$ unchanged. Therefore all the singletons should be inserted into the basis, before selecting the columns on the basis of minimum $D_j^{(k)}$.

There is one factor which we have not taken into consideration so far: the value of element $a_{rs}^{(k)}$ ($a_{rs}^{(k)}$ is known as pivot). Very small pivots should be avoided in consideration of round-off errors and keeping the elements $a_{rs}^{(k)}$ bounded (7). We shall discuss the pivot size from a geometrical viewpoint. The angle between the vectors e_r and $a_s^{(k)}$ is

$$\gamma_r = \cos^{-1}\left(\frac{e_r \cdot a_s(k)}{||e_r|| \cdot ||\, a_s^{(k)}\, ||}\right)$$

where the norms denote the lengths of the vectors. Therefore

$$\gamma_r = \cos^{-1}\left(\frac{a_{rs}^{(k)}}{||a_s^{(k)}||}\right)$$

For γ_r to be small $a_{rs}^{(k)}$ should be large in absolute value. Thus by choosing the largest pivot we transform $a_s^{(k)}$ to the original available basic vector making the smallest angle with $a_s^{(k)}$. Small pivot implies that $a_s^{(k)}$ is nearly orthogonal to unit basic vector corresponding to the position of small pivot. In practice (4), when choosing a pivot, only the rows that have elements greater than a certain tolerance in absolute value are considered. This avoids replacing an original basic vector that is nearly orthogonal to incoming vector.

The computation of $D_j^{(k)}$ ($j = k + 1, \ldots, n$; $k = 0, 1, \ldots, n$) and sorting the column vectors according to ascending values of $D_j^{(k)}$ at every step k, is quite time consuming in practice. Therefore $D_j^{(0)}$ are computed and columns are sorted in ascending order of $D_j^{(0)}$, $j = 1, 2, \ldots, n$, only once at the beginning. The $c^{(k)}$ vector is also obtained by suitably modifying originally computed vector $c^{(0)}$. The present methods of modifying $c^{(k)}$ to $c^{(k+i)}$ (2,4) are as follows. All elements of $c^{(k)}$ that correspond to non-zero elements of the vector entering the basis are decreased by unity when a vector enters the basis. This does not take into consideration the growth in the density of remaining non-basic column vectors and the consequent change in $c^{(k)}$. We have found that approximate probability arguments can be utilized to obtain heuristically an approximate formula for computing $c^{(k+i)}$, $k \neq 0$ as follows. Consider the matrix $A^{(k)}$. There is no loss of generality if we assume that its first k-columns have been already transformed into unit vectors and vector $(k + 1)$ is relpacing original unit vector e_r.

Column No. →	k + 1	k + 2	j	n	$c^{(k)}$
.	.	.	.	.	.
Row i	x	.	$a_{ij}^{(k)}$	.	$c_i^{(k)}$
.	.	.	.	.	.
Row r	x	.	$a_{rj}^{(k)}$	.	$c_r^{(k)}$
.	.	.	.	.	.

Let r be the pivot row and i be any one of the rows of the vector k + 1 that has a non-zero element. Then there are $c_i^{(k)} - 1$ elements in other non-basic columns of row i. Let $k + 2 \leq j \leq n$. If we assume that the non-zero elements in row i are randomly distributed then the probability of $a_{ij}^{(k)}$ being non-zero is approximately given by

$$P\left(a_{ij}^{(k)} \neq 0\right) \approx \frac{c_i^{(k)} - 1}{n-k-1}$$

or

$$P\left(a_{ij}^{(k)} = 0\right) \approx 1 - \frac{c_i^{(k)} - 1}{n-k-1}$$

Similarly

$$P\left(a_{rj}^{(k)} \neq 0\right) \approx \frac{c_r^{(k)} - 1}{n-k-1}$$

If $a_{rj}^{(k)} \neq 0$ and $a_{ij}^{(k)}$ was zero then we have an increase of non-zero elements in row i and hence an increase in the probability of a non-zero $a_{ij}^{(k+1)}$.

Assuming that the events $a_{rj}^{(k)} \neq 0$ and $a_{ij}^{(k)} = 0$ are independent $P(a_{rj}^{(k)} \neq 0 \text{ and } a_{ij}^{(k)} = 0) \approx P(a_{rj}^{(k)} \neq 0)$.

$$P(a_{ij}^{(k)} = 0) \approx \left(\frac{c_r^{(k)} -1}{n-k-1}\right)\left(1 - \frac{c_i^{(k)} -1}{n-k-1}\right)$$

Hence

$$P\left(a_{ij}^{(k+1)} \neq 0\right) = P\left(a_{ij}^{(k)} \neq 0\right) + P\left(a_{ij}^{(k)} = 0 \text{ and beconing non-zero}\right)$$

or

$$\frac{c_i^{(k+1)}}{n-k-1} = \left(\frac{c_i^{(k)} -1}{n-k-1}\right) + \left(\frac{c_r^{(k)} -1}{n-k-1}\right)\left(1 - \frac{c_i^{(k)} -1}{n-k-1}\right)$$

[1] $$c_i^{(k+1)} = c_i^{(k)} + c_r^{(k)} - 2 - \frac{(c_r^{(k)} -1)\ (c_i^{(k)} -1)}{n-k-1}, \quad i \neq r$$

of course,

$$c_r^{(k+1)} = c_r^{(k)} - 1.$$

If the $c_i^{(k+1)}$ computed on the basis of the above formula comes out to be negative, it is set to zero and if it turns out to be greater than $n - k - 1$, we will set it equal to $n - k - 1$. It is evident that the above is easy to incorporate in computer codes where some form of updating of $c^{(k)}$ is already in use.

Numerical results

Twenty-two 50 x 50 matrices were constructed. The non-zero elements of these matrices are randomly distributed. Their densities vary from .027 to .150. The PFI of each of these matrices was determined using four different methods, given below.

I (i) Columns of the A matrix were selected sequentially, beginning with the first column, to enter the basis.

(ii) The original unit basic vectors were removed on observing the minimum row count. The row count vector was updated (modified) in the ordinary way, viz., corresponding to each non-zero element of the vector going into the basis unity was subtracted from the corresponding element of the row count vector.

II Same as I, except the row-count vector was updated (modified) using formulas [1].

III Columns were brought into the basis in the order of ascending values of $D_j^{(0)}$: j = 1, 2, ..., n. and then same as I (ii).

IV Same as III except that the row-count vector was updated using formulas (1).

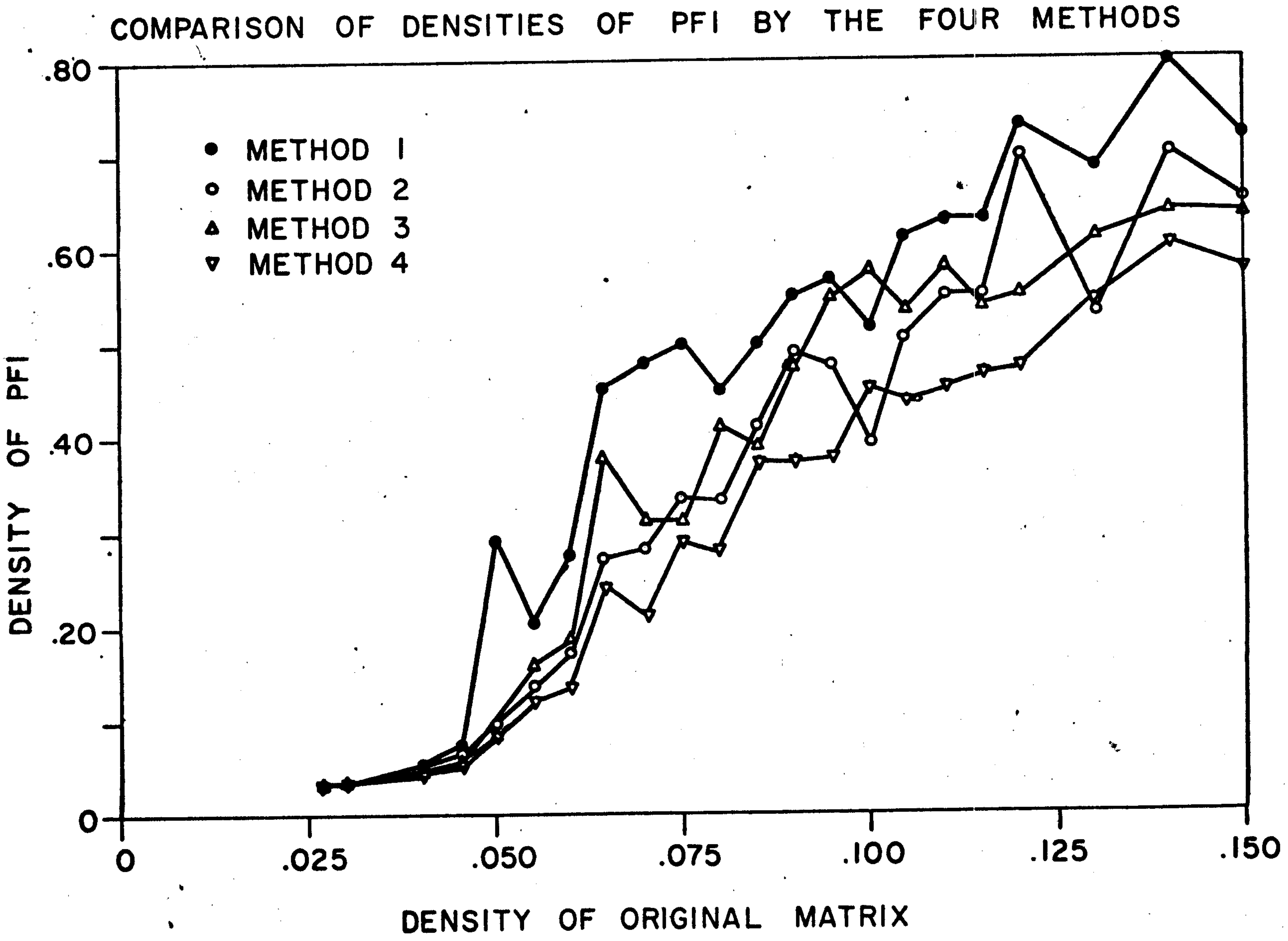
COMPARISON OF DENSITIES OF PFI BY THE FOUR METHODS
● METHOD 1
○ METHOD 2
△ METHOD 3
▽ METHOD 4
DENSITY OF PFI
.80
.60
.40
.20
0
0
.025
.050
.075
.100
.125
.150
DENSITY OF ORIGINAL MATRIX

Fig. 1

The relationships between the densities of the original matrices and their PFI's, for each of the four methods, are shown in Figure 1. The densities of the original matrices are represented along the abscissa and their PFI's along the ordinate. It is clear from the figure that Method II is always better than Method I; viz., for a given matrix the density of its PFI, using Method II, is always less than the density of PFI obtained by Method I. Since the approximate probability arguments utilized for the modification of the row count vector in Method II become increasingly incorrect fór larger values for the density d of the original matrix, some oscillations are observed in the graph of Method II for d > .08. Method III, which involves sorting of the columns in actual linear programming codes and therefore is not only difficult to program, but also slow in operation, gives nearly the same PFI density as Method II. In comparison with method I, Methods II, III and IV, on the average, led to 24%, 19% and 32% fewer non-zero entries respectively. If one is willing to pay the price of sorting the columns then Method IV is recommended since it seems to be the best of all.

Acknowledgement: The programming assistance of G. Krishna Rao and E. Koh is gratefully acknowledged.

REFERENCES

1. Dantzig, G. B. and Orch ard-Hays, W.: The Product Form of the Inverse in the Simplex Method. MTAC, 8:46 (April) 1954, pp. 64-67.

2. Smith, D. M. and Orchard-Hays, W.: Computational Efficiency in Product Form L. P. Codes. Recent Advances in Mathematical Programming. Graves, R. L. and Wolfe, P. Eds. McGraw-Hill, New York, 1963, pp. 211-218.

3. Cutler, L. and Wolfe, P.: Experiments in Linear Programming. Ibid, pp. 177-200.

4. Larson, L. J.: A Modified Inversion Procedure for Product Form of Inverse in Linear Programming Codes. Comm. of ACM, (July) 1962, pp. 382-83.

5. Markowitz, H. M.: The Elimination Form of the Inverse and its Applications to Linear Programming. Management Science, 3 (April) 1957, pp. 255-69.

6. Hildebrand, F. B.: Introduction to Numerical Analysis. McGraw-Hill, New York, 1956, pp.429.

7. Wilkinson, J. H.: Reounding Errors in Algebraic Processes. Prentice-Hall, Englewood Cliffs, N. J., 1963, pp. 97.